Interactive verification of Markov chains: Two distributed protocol case studies

Johannes Hölzl and Tobias Nipkow

TU München

QFM 2012
28 August 2012

Introduction

- In the interactive theorem prover

Isabelle

Introduction

- In the interactive theorem prover

Isabelle

- Verified two case studies

Introduction

- In the interactive theorem prover

Isabelle

- Verified two case studies
  - ZeroConf protocol (IPv4 address allocation)

Introduction

- In the interactive theorem prover

Isabelle

- Verified two case studies
  - ZeroConf protocol (IPv4 address allocation)
  - Crowds protocol (anonymizing service)

Introduction

- In the interactive theorem prover

Isabelle

- Verified two case studies
  - ZeroConf protocol (IPv4 address allocation)
  - Crowds protocol (anonymizing service)
- Built on Isabelle's probability theory and Markov chains
  Hölzl & Heller (ITP 2011), Hölzl & Nipkow (TACAS 2012)

Interactive theorem proving

Interactive theorem proving

- Mathematics, but checked by a computer

Interactive theorem proving

- Mathematics, but checked by a computer
- Powerful logics (e.g. ZF, CoC, HOL):

Interactive theorem proving

- Mathematics, but checked by a computer
- Powerful logics (e.g. ZF, CoC, HOL):
  - Can deal with infinite-state systems

Interactive theorem proving

- Mathematics, but checked by a computer
- Powerful logics (e.g. ZF, CoC, HOL):
  - Can deal with infinite-state systems
  - User-extensible

Interactive theorem proving

- Mathematics, but checked by a computer
- Powerful logics (e.g. ZF, CoC, HOL):
  - Can deal with infinite-state systems
  - User-extensible
- Too powerful to be fully automatic:

Interactive theorem proving

- Mathematics, but checked by a computer
- Powerful logics (e.g. ZF, CoC, HOL):
  - Can deal with infinite-state systems
  - User-extensible
- Too powerful to be fully automatic:
  user needs to write proofs

Interactive theorem proving

- Mathematics, but checked by a computer
- Powerful logics (e.g. ZF, CoC, HOL):
  - Can deal with infinite-state systems
  - User-extensible
- Too powerful to be fully automatic:
  user needs to write proofs
- Proof language and proof methods

Isabelle/HOL

- Logic is HOL: functional programming + quantifiers

Isabelle/HOL

- Logic is HOL: functional programming + quantifiers
- Declarative proof language Isar

Isabelle/HOL

- Logic is HOL: functional programming + quantifiers
- Declarative proof language Isar
- Small kernel: each proof is reduced to primitive proof steps

Isabelle/HOL

- Logic is HOL: functional programming + quantifiers
- Declarative proof language Isar
- Small kernel: each proof is reduced to primitive proof steps
- Powerful proof methods
  (rewrite engine, Sledgehammer, ...)

Isabelle/HOL

- Logic is HOL: functional programming + quantifiers
- Declarative proof language Isar
- Small kernel: each proof is reduced to primitive proof steps
- Powerful proof methods
  (rewrite engine, Sledgehammer, ...)
- Important theories: datatypes, real analysis, measure theory, probability theory, Markov chains, ...

Case study: ZeroConf protocol

ZeroConf protocol

- Protocol to allocate an address in a link-local network, without central authority (RFC 3927)

ZeroConf protocol

- Protocol to allocate an address in a link-local network, without central authority (RFC 3927)
- We formalized the analysis of Bohnenkamp *et al.* (2003)

ZeroConf protocol

- Protocol to allocate an address in a link-local network, without central authority (RFC 3927)
- We formalized the analysis of Bohnenkamp *et al.* (2003)
  - Address allocation when only one computer is added

ZeroConf protocol

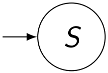
- Protocol to allocate an address in a link-local network, without central authority (RFC 3927)
- We formalized the analysis of Bohnenkamp *et al.* (2003)
  - Address allocation when only one computer is added
  - Probability that two hosts end up with the same address

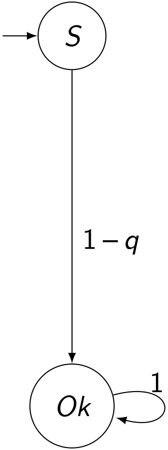
ZeroConf protocol

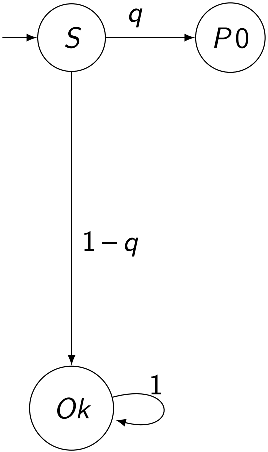
- Protocol to allocate an address in a link-local network, without central authority (RFC 3927)
- We formalized the analysis of Bohnenkamp *et al.* (2003)
  - Address allocation when only one computer is added
  - Probability that two hosts end up with the same address
  - Expected time until an address is allocated

ZeroConf protocol

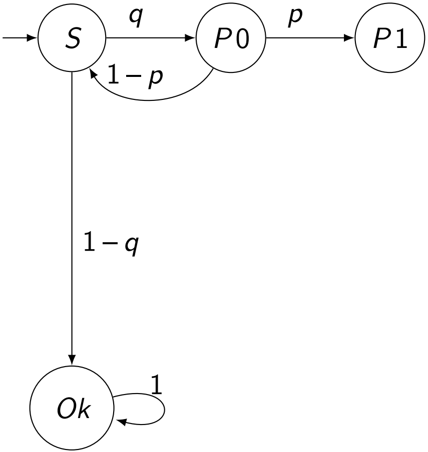
- Protocol to allocate an address in a link-local network, without central authority (RFC 3927)
- We formalized the analysis of Bohnenkamp *et al.* (2003)
  - Address allocation when only one computer is added
  - Probability that two hosts end up with the same address
  - Expected time until an address is allocated
- Model checking analysis of Kwiatkowska *et al.* (2006) and Andova *et al.* (2003)

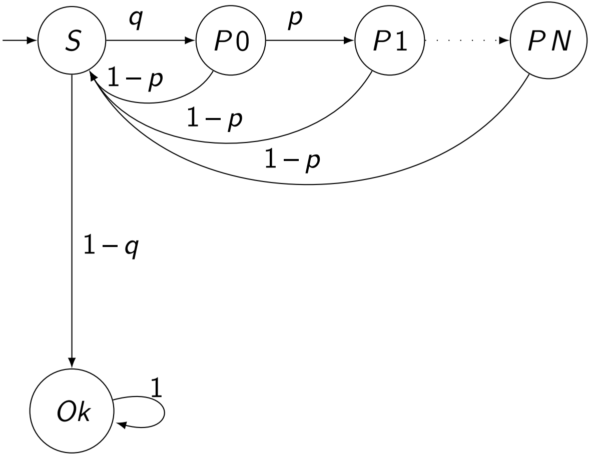

S


S
1 − q
Ok
1

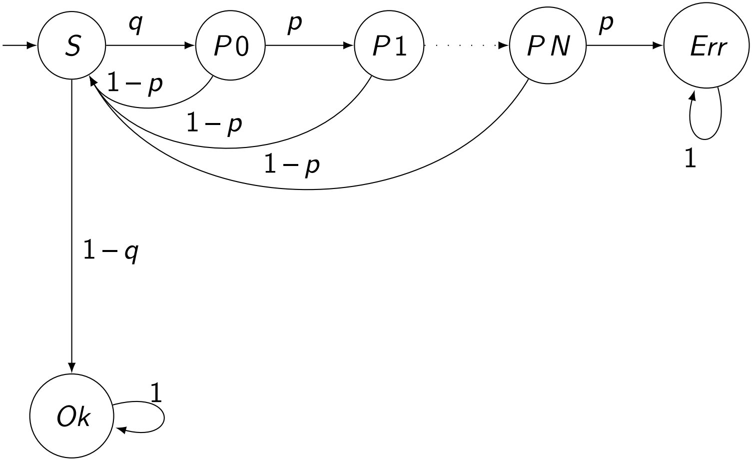


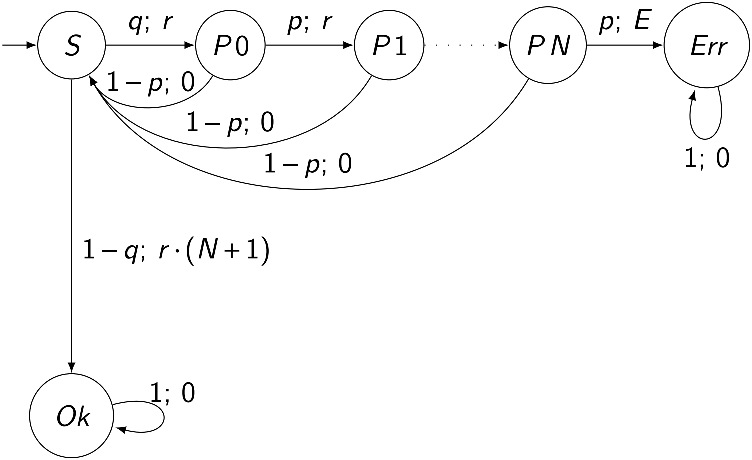
q
S
$P0$
$1-q$
Ok
1


S
P0
P1
Ok
q
p
1 − p
1 − q
1



S
q
P0
p
P1
PN
1 − p
1 − p
1 − p
1 − q
Ok
1


S
q
P 0
p
P 1
P N
p
Err
1 − p
1 − p
1 − p
1
1 − q
Ok
1


S
q; r
P0
p; r
P1
PN
p; E
Err
1 − p; 0
1 − p; 0
1 − p; 0
1; 0
1 − q; r · (N + 1)
Ok
1; 0

- Fix parameters:

  fixes $N :: \mathbb{N}$ **and** $p\ q\ r\ E :: \mathbb{R}$

- Fix parameters:

$$\textbf{fixes } N :: \mathbb{N} \textbf{ and } p\ q\ r\ E :: \mathbb{R}$$

- Fix parameters:

fixes $N :: \mathbb{N}$ **and** $p\ q\ r\ E :: \mathbb{R}$
assumes $0 < p$ **and** $p < 1$ **and** $0 < q$ **and** $q < 1$

- Fix parameters:

fixes $N :: \mathbb{N}$ **and** $p\ q\ r\ E :: \mathbb{R}$
assumes $0 < p$ **and** $p < 1$ **and** $0 < q$ **and** $q < 1$
assumes $0 \leq E$ **and** $0 \leq r$

- Fix parameters:

$$\textbf{fixes}\ N :: \mathbb{N}\ \textbf{and}\ p\ q\ r\ E :: \mathbb{R}$$
$$\textbf{assumes}\ 0 < p\ \textbf{and}\ p < 1\ \textbf{and}\ 0 < q\ \textbf{and}\ q < 1$$
$$\textbf{assumes}\ 0 \le E\ \textbf{and}\ 0 \le r$$

- Define state space:

$$\textbf{datatype}\ \textit{zc-state} = S \mid P\ \mathbb{N} \mid Ok \mid Err$$

$$\Omega = \left\{S, Ok, Err\right\} \cup \left\{P\,n \;\middle|\; n \le N\right\}$$

- Fix parameters:

$$\textbf{fixes}\ N :: \mathbb{N}\ \textbf{and}\ p\ q\ r\ E :: \mathbb{R}$$
$$\textbf{assumes}\ 0 < p\ \textbf{and}\ p < 1\ \textbf{and}\ 0 < q\ \textbf{and}\ q < 1$$
$$\textbf{assumes}\ 0 \le E\ \textbf{and}\ 0 \le r$$

- Define state space:

$$\textbf{datatype}\ \textit{zc-state} = S \mid P\ \mathbb{N} \mid Ok \mid Err$$

$$\Omega = \left\{S, Ok, Err\right\} \cup \left\{P\,n \;\middle|\; n \le N\right\}$$

- Define the transition function τ:

$$\begin{array}{llcl} \tau\ S & Ok & = & 1 - q \\ \tau\ S & (P\ 0) & = & q \\ \tau\ (P\,n) & (P\,(n+1)) & = & \text{if } n < N \text{ then } p \text{ else } 0 \\ & & \vdots & \end{array}$$

- Defines a Markov chain:

$$\textbf{lemma} \quad \textit{markov-chain}\ \Omega\ \tau$$

- Defines a Markov chain:

lemma *markov-chain* $\Omega\ \tau$

- Probability theory gives us:

  $\text{Pr}_s(\omega.\ P\,\omega)$ – the probability that a trace ω fulfills $P\,\omega$

- Defines a Markov chain:

$$\textbf{lemma} \quad \textit{markov-chain}\ \Omega\ \tau$$

- Probability theory gives us:

  $\mathrm{Pr}_s(\omega.\ P\,\omega)$ – the probability that a trace ω fulfills $P\,\omega$

- Define probability that an error is reached:

$$\mathrm{P}_{\mathrm{err}}\ s = \mathrm{Pr}_s(\omega.\ \exists n.\ \omega\ n = \mathit{Err})$$

- Defines a Markov chain:

$$\textbf{lemma} \quad \textit{markov-chain}\ \Omega\ \tau$$

- Probability theory gives us:

  $\mathrm{Pr}_s(\omega.\ P\,\omega)$ – the probability that a trace ω fulfills $P\,\omega$

- Define probability that an error is reached:

$$\mathrm{P_{err}}\ s = \mathrm{Pr}_s(\omega.\ \exists n.\ \omega\ n = \mathit{Err})$$

- Analyse: $\mathrm{P_{err}}\ S = ?$

lemma

$$n \le N \implies \mathsf{P}_{\mathsf{err}}\ (P\,(N-n)) = p^{n+1} + (1-p^{n+1}) \cdot \mathsf{P}_{\mathsf{err}}\ S$$

lemma

$$n \leq N \Longrightarrow \mathsf{P}_{\mathsf{err}}\ (P\,(N - n)) = p^{n+1} + (1 - p^{n+1}) \cdot \mathsf{P}_{\mathsf{err}}\ S$$

proof (*induct n*)

lemma

$$n \leq N \implies \mathsf{P}_{\mathsf{err}}\ (P\,(N-n)) = p^{n+1} + (1-p^{n+1}) \cdot \mathsf{P}_{\mathsf{err}}\ S$$

proof (*induct n*)

case $(n+1)$

lemma

$$n \le N \Longrightarrow \mathsf{P}_{\mathsf{err}}\ (P\,(N-n)) = p^{n+1} + (1-p^{n+1}) \cdot \mathsf{P}_{\mathsf{err}}\ S$$

proof (*induct n*)

case $(n+1)$

have $\mathsf{P}_{\mathsf{err}}\ (P\,(N-(n+1)))$

$$= p \cdot (p^{n+1} + (1-p^{n+1}) \cdot \mathsf{P}_{\mathsf{err}}\ S) + (1-p) \cdot \mathsf{P}_{\mathsf{err}}\ S$$

by (*simp*$\cdots$)

lemma

$$n \le N \Longrightarrow \mathrm{P_{err}}\ (P\,(N - n)) = p^{n+1} + (1 - p^{n+1}) \cdot \mathrm{P_{err}}\ S$$

proof (*induct n*)

case $(n+1)$

have $\mathrm{P_{err}}\ (P\,(N - (n+1)))$

$$= p \cdot (p^{n+1} + (1 - p^{n+1}) \cdot \mathrm{P_{err}}\ S) + (1 - p) \cdot \mathrm{P_{err}}\ S$$

by (*simp* $\cdots$)

also have $\ldots = p^{(n+1)+1} + (1 - p^{(n+1)+1}) \cdot \mathrm{P_{err}}\ S$

by (*simp* $\cdots$)

lemma

$$n \le N \Longrightarrow \mathrm{P}_{\mathrm{err}}\ (P\,(N - n)) = p^{n+1} + (1 - p^{n+1}) \cdot \mathrm{P}_{\mathrm{err}}\ S$$

proof $(\mathit{induct}\ n)$

case $(n+1)$

have $\mathrm{P}_{\mathrm{err}}\ (P\,(N - (n+1)))$

$$= p \cdot (p^{n+1} + (1 - p^{n+1}) \cdot \mathrm{P}_{\mathrm{err}}\ S) + (1 - p) \cdot \mathrm{P}_{\mathrm{err}}\ S$$

by $(\mathit{simp} \cdots)$

also have $\ldots = p^{(n+1)+1} + (1 - p^{(n+1)+1}) \cdot \mathrm{P}_{\mathrm{err}}\ S$

by $(\mathit{simp} \cdots)$

finally show $\mathrm{P}_{\mathrm{err}}\ (P\,(N - (n+1)))$

$$= p^{(n+1)+1} + (1 - p^{(n+1)+1}) \cdot \mathrm{P}_{\mathrm{err}}\ S\ .$$

next

lemma

$$n \le N \Longrightarrow \mathrm{P_{err}}\ (P\,(N-n)) = p^{n+1} + (1-p^{n+1}) \cdot \mathrm{P_{err}}\ S$$

proof (*induct n*)

case $(n+1)$

have $\mathrm{P_{err}}\ (P\,(N-(n+1)))$

$$= p \cdot (p^{n+1} + (1-p^{n+1}) \cdot \mathrm{P_{err}}\ S) + (1-p) \cdot \mathrm{P_{err}}\ S$$

by (*simp*$\cdots$)

also have $\ldots = p^{(n+1)+1} + (1-p^{(n+1)+1}) \cdot \mathrm{P_{err}}\ S$

by (*simp*$\cdots$)

finally show $\mathrm{P_{err}}\ (P\,(N-(n+1)))$

$$= p^{(n+1)+1} + (1-p^{(n+1)+1}) \cdot \mathrm{P_{err}}\ S\ .$$

next

case 0

show $\mathrm{P_{err}}\ (P\,(N-0)) = p^{0+1} + (1-p^{0+1}) \cdot \mathrm{P_{err}}\ S$

by *simp*

lemma

$$n \le N \Longrightarrow \mathsf{P_{err}}\ (P\,(N - n)) = p^{n+1} + (1 - p^{n+1}) \cdot \mathsf{P_{err}}\ S$$

proof (*induct n*)

case $(n+1)$

have $\mathsf{P_{err}}\ (P\,(N - (n+1)))$

$= p \cdot (p^{n+1} + (1 - p^{n+1}) \cdot \mathsf{P_{err}}\ S) + (1 - p) \cdot \mathsf{P_{err}}\ S$

by $(simp \cdots)$

also have $\ldots = p^{(n+1)+1} + (1 - p^{(n+1)+1}) \cdot \mathsf{P_{err}}\ S$

by $(simp \cdots)$

finally show $\mathsf{P_{err}}\ (P\,(N - (n+1)))$

$= p^{(n+1)+1} + (1 - p^{(n+1)+1}) \cdot \mathsf{P_{err}}\ S$.

next

case 0

show $\mathsf{P_{err}}\ (P\,(N - 0)) = p^{0+1} + (1 - p^{0+1}) \cdot \mathsf{P_{err}}\ S$

by *simp*

qed

- General result:

theorem $$\mathsf{P}_{\mathsf{err}}\ S = \frac{q \cdot p^{N+1}}{1 - q \cdot (1 - p^{N+1})}$$

- General result:

theorem $$\mathrm{P_{err}}\ S = \frac{q \cdot p^{N+1}}{1 - q \cdot (1 - p^{N+1})}$$

- 16 hosts ($q = 16/65024$), 3 probe runs ($N = 2$), $p = 0.01$:

corollary $$\mathrm{P_{err}}\ S \leq 10^{-13}$$

- How do we model the expected running time?

- How do we model the expected running time?
- Similar to τ define the cost function ρ:

$$\begin{array}{lllll} \rho & S & Ok & = & r \cdot (N+1) \\ \rho & S & (P\,0) & = & r \\ \rho & (P\,n) & (P\,(n+1)) & = & \text{if } n < N \text{ then } r \text{ else } 0 \\ & & & \vdots & \end{array}$$

- How do we model the expected running time?
- Similar to τ define the cost function ρ:

$$\begin{array}{llcl} \rho\ S & Ok & = & r \cdot (N+1) \\ \rho\ S & (P\,0) & = & r \\ \rho\ (P\,n) & (P\,(n+1)) & = & \text{if } n < N \text{ then } r \text{ else } 0 \\ & & \vdots & \end{array}$$

- Define expected cost until *Err* or *Ok* is reached:

$$\mathsf{C}_{\mathsf{fin}}\ s = \int_{\omega} \mathit{cost\text{-}until}\ \Big\{ \mathit{Err}, \mathit{Ok} \Big\}\ (s \cdot \omega)\ \mathsf{dPr}_s$$

- How do we model the expected running time?
- Similar to τ define the cost function ρ:

$$
\begin{array}{lll}
\rho\ S\ \ Ok & = & r \cdot (N+1) \\
\rho\ S\ \ (P\,0) & = & r \\
\rho\ (P\,n)\ (P\,(n+1)) & = & \text{if } n < N \text{ then } r \text{ else } 0 \\
& \vdots &
\end{array}
$$

- Define expected cost until *Err* or *Ok* is reached:

$$\mathrm{C_{fin}}\ s = \int_{\omega} \mathit{cost\text{-}until}\ \{Err, Ok\}\ (s \cdot \omega)\ \mathrm{dPr}_s$$

- 16 hosts, 3 probe runs, $p = 0.01$, $r = 2ms$, $E = 3600s$:

theorem $\mathrm{C_{fin}}\ S \leq 0.007$

Case study: Crowds protocol

Crowds protocol

- Anonymizing protocol
  introduced and analysed by Reiter & Rubin (1998)

Crowds protocol

- Anonymizing protocol
  introduced and analysed by Reiter & Rubin (1998)
- Group of *nodes* establishes a connection by randomly chosing another node or the final server

Crowds protocol

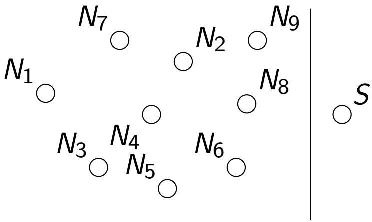
- Anonymizing protocol
  introduced and analysed by Reiter & Rubin (1998)
- Group of *nodes* establishes a connection by randomly chosing another node or the final server
- Analysis:

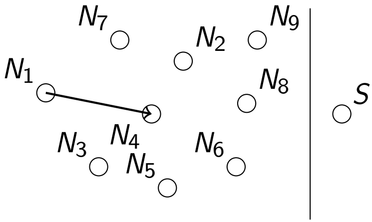
Crowds protocol

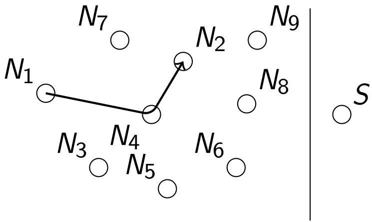
- Anonymizing protocol
  introduced and analysed by Reiter & Rubin (1998)
- Group of *nodes* establishes a connection by randomly chosing another node or the final server
- Analysis:
  - Probability that original sender contacts a collaborating node is small

Crowds protocol

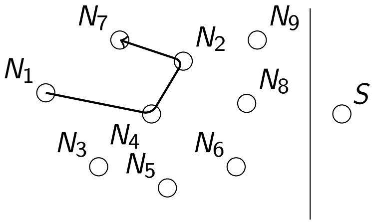
- Anonymizing protocol
  introduced and analysed by Reiter & Rubin (1998)
- Group of *nodes* establishes a connection by randomly chosing another node or the final server
- Analysis:
  - Probability that original sender contacts a collaborating node is small
  - Information a contacted collaborating node gains is small

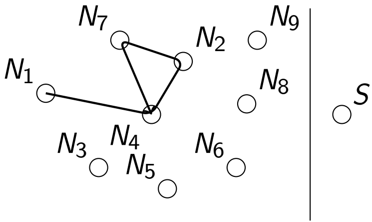

N_7
N_9
N_2
N_1
N_8
S
N_4
N_3
N_6
N_5

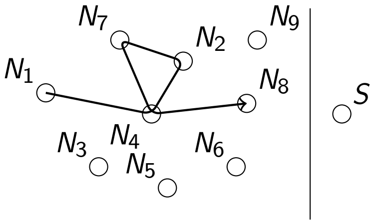

N_7
N_2
N_9
N_1
N_8
S
N_4
N_3
N_6
N_5

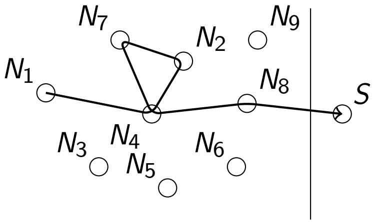


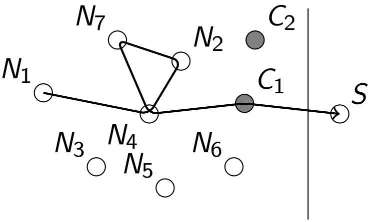
N_7
N_9
N_2
N_1
N_8
S
N_4
N_3
N_6
N_5


N_7
N_2
N_9
N_1
N_8
S
N_4
N_3
N_5
N_6

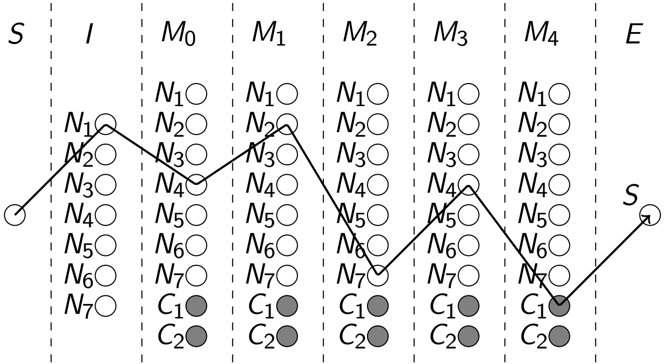

N_7
N_2
N_9
N_1
N_8
S
N_4
N_3
N_6
N_5


N_7
N_9
N_2
N_1
N_8
S
N_4
N_3
N_6
N_5

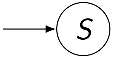

N_7
N_9
N_2
N_1
N_8
S
N_3
N_4
N_5
N_6



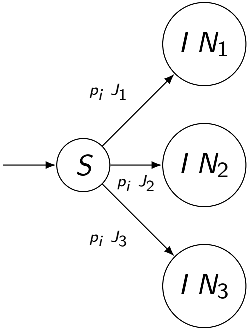
N_7
C_2
N_2
N_1
C_1
S
N_4
N_3
N_6
N_5


S
I
M_0
M_1
M_2
M_3
M_4
E
N_1 N_2 N_3 N_4 N_5 N_6 N_7
C_1 C_2
S

Probabilities:

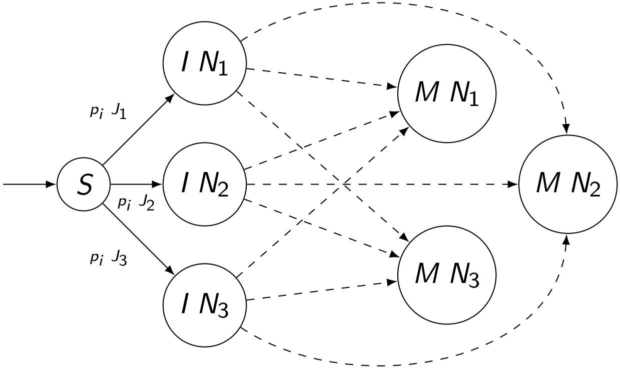


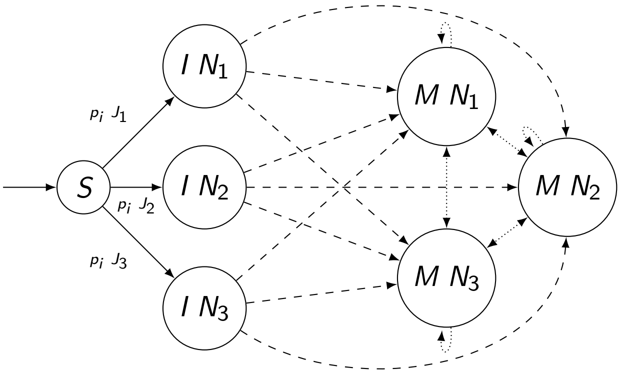
Probabilities:



Probabilities:

1/3

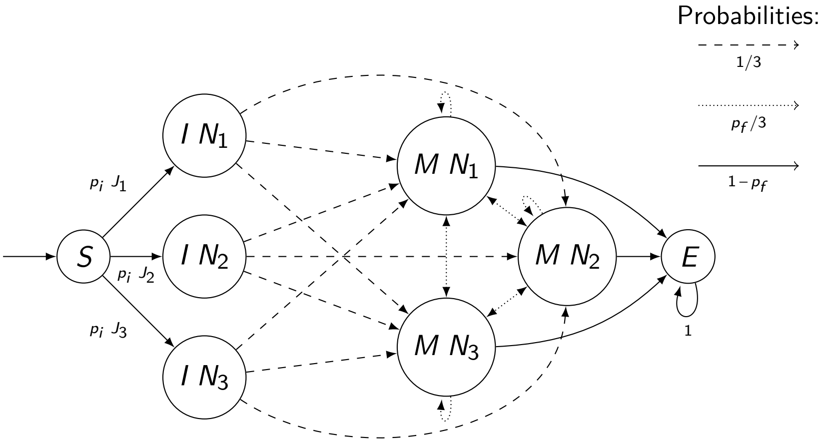




Probabilities:

1/3

$p_f/3$


Probabilities:
1/3
$p_f/3$
$1-p_f$
S
$I\ N_1$
$I\ N_2$
$I\ N_3$
$M\ N_1$
$M\ N_2$
$M\ N_3$
E
$p_i\ J_1$
$p_i\ J_2$
$p_i\ J_3$
1

- Fix parameters

$$\textbf{fixes}\ N\ C :: \textit{node set}\ \textbf{and}\ p_f :: \mathbb{R}\ \textbf{and}\ p_i :: \textit{node} \rightarrow \mathbb{R}$$

- Fix parameters

$$\textbf{fixes}\ N\ C :: \textit{node set}\ \textbf{and}\ p_f :: \mathbb{R}\ \textbf{and}\ p_i :: \textit{node} \rightarrow \mathbb{R}$$

- Fix parameters

  fixes $N\ C :: \mathit{node\ set}$ **and** $p_f :: \mathbb{R}$ **and** $p_i :: \mathit{node} \rightarrow \mathbb{R}$
  assumes $0 < p_f$ **and** $p_f < 1$

- Fix parameters

  fixes $N\ C :: \mathit{node\ set}$ **and** $p_f :: \mathbb{R}$ **and** $p_i :: \mathit{node} \rightarrow \mathbb{R}$
  assumes $0 < p_f$ **and** $p_f < 1$
  assumes $N \neq \emptyset$ **and** $\mathit{finite}\ N$

- Fix parameters

fixes $N\ C :: node\ set$ **and** $p_f :: \mathbb{R}$ **and** $p_i :: node \rightarrow \mathbb{R}$
assumes $0 < p_f$ **and** $p_f < 1$
assumes $N \neq \emptyset$ **and** $finite\ N$
assumes $\forall n \in N.\ 0 \leq p_i\ n$ **and** $\sum_{n \in N} p_i\ j = 1$

- Fix parameters

fixes $N\ C :: node\ set$ **and** $p_f :: \mathbb{R}$ **and** $p_i :: node \rightarrow \mathbb{R}$
assumes $0 < p_f$ **and** $p_f < 1$
assumes $N \neq \emptyset$ **and** $finite\ N$
assumes $\forall n \in N.\ 0 \leq p_i\ n$ **and** $\sum_{n \in N} p_i\ j = 1$
assumes $C \neq \emptyset$ **and** $C \subset N$ **and** $\forall c \in C.\ p_i\ c = 0$

- Fix parameters

$$\textbf{fixes } N\ C :: \textit{node set } \textbf{and } p_f :: \mathbb{R} \textbf{ and } p_i :: \textit{node} \rightarrow \mathbb{R}$$
$$\textbf{assumes } 0 < p_f \textbf{ and } p_f < 1$$
$$\textbf{assumes } N \neq \emptyset \textbf{ and } \textit{finite } N$$
$$\textbf{assumes } \forall n \in N.\ 0 \leq p_i\ n \textbf{ and } \textstyle\sum_{n \in N} p_i\ j = 1$$
$$\textbf{assumes } C \neq \emptyset \textbf{ and } C \subset N \textbf{ and } \forall c \in C.\ p_i\ c = 0$$

- Define state space

$$\textbf{datatype } \alpha\ \textit{c-state} = S \mid I\ \alpha \mid M\ \alpha \mid E$$

$$\Omega = \left\{S\right\} \cup \left\{I\ n \,\middle|\, n \in N \setminus C\right\} \cup \left\{M\ n \,\middle|\, n \in N\right\} \cup \left\{E\right\}$$

- Define transition function

$$
\begin{aligned}
\tau\ S &\quad (I\ n) &&= p_i\ n \\
\tau\ (I\ n) &\quad (M\ n') &&= 1/|N| \\
\tau\ (M\ n) &\quad (M\ n') &&= p_f/|N| \\
\tau\ (M\ n) &\quad E &&= 1 - p_f \\
\tau\ E &\quad E &&= 1 \\
\tau\ \_ &\quad \_ &&= 0
\end{aligned}
$$

- Define transition function

$$
\begin{array}{lll}
\tau\ S & (I\ n) & = p_i\ n \\
\tau\ (I\ n) & (M\ n') & = 1/|N| \\
\tau\ (M\ n) & (M\ n') & = p_f/|N| \\
\tau\ (M\ n) & E & = 1 - p_f \\
\tau\ E & E & = 1 \\
\tau\ \_ & \_ & = 0
\end{array}
$$

- Prove Markov chain property

theorem *markov-chain* $\Omega\ \tau$

- We introduce some random variables:

- We introduce some random variables:

- We introduce some random variables:

  init the initiating node

- We introduce some random variables:

init	the initiating node
last-ncoll	the first node contacting a collaborating node

- We introduce some random variables:

init	the initiating node
last-ncoll	the first node contacting a collaborating node
hit	true if a collaborating node is contacted

- We introduce some random variables:

init	the initiating node
last-ncoll	the first node contacting a collaborating node
hit	true if a collaborating node is contacted

- Probability that initiating node contacts a collaborating node

theorem $\Pr_S(\omega.\ \mathit{init}\ \omega = \mathit{last\text{-}ncoll}\ \omega \mid \mathit{hit}\ \omega) = 1 - \frac{|N \setminus C| - 1}{|N|} \cdot p_f$

- We introduce some random variables:

init	the initiating node
last-ncoll	the first node contacting a collaborating node
hit	true if a collaborating node is contacted

- Probability that initiating node contacts a collaborating node

theorem $\Pr_S(\omega.\, \mathit{init}\ \omega = \mathit{last\text{-}ncoll}\ \omega \mid \mathit{hit}\ \omega) = 1 - \frac{|N \setminus C| - 1}{|N|} \cdot p_f$

- Information the collaborating nodes gain when contacted

theorem $I_{\mathit{hit}}(\mathit{init}; \mathit{last\text{-}ncoll}) \leq \left(1 - \frac{|N \setminus C| - 1}{|N|} \cdot p_f\right) \cdot \log_2 |N \setminus C|$

Related Work: probability theory in ITPs

- Probability space of boolean sequences: $\mathbb{N} \rightarrow \{0,1\}$
  Hurd (2002), Hasan *et al.* (2009), Liu *et al.* (2011)

Related Work: probability theory in ITPs

- Probability space of boolean sequences: $\mathbb{N} \to \{0, 1\}$
  Hurd (2002), Hasan *et al.* (2009), Liu *et al.* (2011)

- Expectation and information theory (discrete, finite spaces)
  Coble (2009)

Related Work: probability theory in ITPs

- Probability space of boolean sequences: $\mathbb{N} \rightarrow \{0,1\}$
  Hurd (2002), Hasan *et al.* (2009), Liu *et al.* (2011)

- Expectation and information theory (discrete, finite spaces)
  Coble (2009)

- Formalization of pGCL (prob. & non-det. language)
  Hurd *et al.* (2005), Audebaud & Paulin-Mohring (2009)

Summary & Future Work

- Markov chains with probability, expectation, and information

Summary & Future Work

- Markov chains with probability, expectation, and information
- ZeroConf protocol: a few days; $\approx$ 300 lines of theory

Summary & Future Work

- Markov chains with probability, expectation, and information
- ZeroConf protocol: a few days; ≈ 300 lines of theory
- Crowds anonymity: a few weeks; $\approx 1,100$ lines of theory

Summary & Future Work

- Markov chains with probability, expectation, and information
- ZeroConf protocol: a few days; ≈ 300 lines of theory
- Crowds anonymity: a few weeks; $\approx 1{,}100$ lines of theory
- Compare: $\approx 20{,}600$ lines of theory for probability theory

Summary & Future Work

- Markov chains with probability, expectation, and information
- ZeroConf protocol: a few days; ≈ 300 lines of theory
- Crowds anonymity: a few weeks; $\approx 1{,}100$ lines of theory
- Compare: $\approx 20{,}600$ lines of theory for probability theory

Future Work:

- More Markov models (MDPs, CTMCs, CTMDPs, PTAs)
- Certification of probabilistic model checker results
- Specification language

Summary & Future Work

- Markov chains with probability, expectation, and information
- ZeroConf protocol: a few days; ≈ 300 lines of theory
- Crowds anonymity: a few weeks; $\approx 1{,}100$ lines of theory
- Compare: $\approx 20{,}600$ lines of theory for probability theory

Future Work:

- More Markov models (MDPs, CTMCs, CTMDPs, PTAs)
- Certification of probabilistic model checker results
- Specification language

Slides available at: `http://www.in.tum.de/~hoelzl`